

Алгоритм нечеткого поиска в базах данных и его практическая реализация

Н.И. Лиманова^а, М.Н. Седов^а

^а Поволжский государственный университет телекоммуникаций и информатики, 443010, ул. Л. Толстого, 23, Самара, Россия

Аннотация

При передаче данных от одного учреждения к другому возникает проблема персональной идентификации физических лиц, у которых частично или полностью не совпадают реквизиты. В работе предложен алгоритм нечеткого поиска, использующий модифицированную метрику Левенштейна и представленный в виде процесса Data Mining. Алгоритм позволяет выполнять поиск физических лиц в базах данных на основе нечёткого сравнения. Он реализован на языке PL-SQL в СУБД Oracle 11g и успешно работает с 2007 года в муниципальных и государственных учреждениях г. Тольятти Самарской области.

Ключевые слова: межведомственный информационный обмен; нечеткое сравнение; поиск персональных данных; функция интеллектуального сравнения

1. Введение

В процессе обработки информации о физических лицах в базах данных для удобства обработки каждому набору реквизитов физических лиц (таких как ФИО, адрес, номера паспорта, СНИЛС и т.п.) присваивается так называемый персональный идентификационный номер (ПИН). В случае обработки или передачи данных о физическом лице вся привязка осуществляется именно к этому ПИНу. При осуществлении обмена информацией о физических лицах между различными учреждениями возникает проблема сопоставления реквизитов из одной базы данных реквизитам в другой. Для однозначной привязки необходимо выполнять нечеткий поиск физического лица в базе-приёмнике, который должен учитывать множество факторов: и потенциальные ошибки при ручном вводе, и отсутствующие или устаревшие реквизиты и т.п. Подобный поиск целесообразно реализовать в виде метода нечеткого поиска и основанного на нем специализированного программного обеспечения [1].

2. Известные методы нечеткого поиска строк

Рассмотрим существующие алгоритмы нечёткого поиска и проанализируем их. Начнём с разработанного Робертом Расселом (Robert C. Russel) и Маргарет Кинг Оделл (Margaret King Odell) алгоритма Soundex [2, 3]. Это один из алгоритмов сравнения двух строк по их звучанию. Он устанавливает одинаковый индекс для строк, имеющих схожее звучание в языке согласно заданной таблице схожих по звучанию символов и их сочетаний. Однако, он имеет существенный недостаток: данный алгоритм привязан к языку, на котором написаны анализируемые строки. Алгоритм используется в настоящее время в основном в англоговорящей среде, для которой подобная таблица уже существует.

Следующий из рассматриваемых алгоритмов — алгоритм расширения выборки [4]. Данный алгоритм часто применяется в системах проверки орфографии. Он основан на сведении задачи о нечетком поиске к задаче о точном поиске. Данный метод подразумевает построение наиболее вероятных «неправильных» вариантов поискового шаблона. Основное достоинство данного алгоритма заключается в легкости его модификации для генерации «ошибочных» вариантов по произвольным правилам. У алгоритма есть и недостатки, главный из которых — большое число проверок для слов существенной длины, поскольку из них можно получить много «ошибочных» слов.

Широко известен алгоритм на основе кода Хэмминга, который применяется при кодировании и декодировании данных. Линейные коды, как правило, хорошо справляются с редкими и большими опечатками. Однако, их эффективность при сравнении слов с частыми, но небольшими ошибками достаточно низкая. В данном алгоритме также присутствуют дополнительные затраты на кодирование информации.

Следующий из рассматриваемых алгоритмов не совсем подходит под поставленную задачу, но для полноты картины не упомянуть о нём всё же нельзя. Это алгоритм, использующий триангуляционные деревья, которые позволяют индексировать множества произвольной структуры при условии, что на них задана метрика. Существует довольно много различных модификаций данного метода, но все они не слишком эффективны в случае текстового поиска и чаще используются в базе данных изображений или других сложных объектов.

Алгоритм Bitap (также известный как Shift-Or или Baeza-Yates-Gonnet) и различные его модификации наиболее часто используются для нечеткого поиска без индексации [5]. Впервые идею этого алгоритма предложили Ricardo Baeza-Yates и Gaston Gonnet, опубликовав соответствующую статью в 1992 году. Оригинальная версия алгоритма имеет дело только с заменами символов, и, фактически, вычисляет расстояние Хемминга. Но немного позже Sun Wu и Udi Manber предложили модификацию этого алгоритма для вычисления расстояния Левенштейна, т.е. привнесли поддержку вставок и удалений и разработали на его основе первую версию утилиты Unix - агрег. Высокая скорость работы этого алгоритма обеспечивается за счет битового параллелизма вычислений: за одну операцию возможно

провести вычисления над 32 и более битами одновременно. При этом, его тривиальная реализация поддерживает поиск слов длиной не более 32 символов. Использование типов больших размерностей замедляет работу алгоритма.

Рассмотрим далее алгоритм Вагнера-Фишера [6], который позволяет для двух строк найти расстояние Левенштейна — минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую. Данный алгоритм имеет ряд значительных преимуществ перед всеми описанными выше, а именно: относительно невысокую сложность реализации, возможность качественного сравнения схожести более чем двух строк, несколько вариантов реализации, которые можно использовать в зависимости от конфигурации системы, универсальность для всевозможных алфавитов. Также у данного алгоритма существует одна интересная модификация, которая позволяет находить расстояние Дамерау-Левенштейна [7]. В нём к операциям вставки, удаления и замены символов, определенных в расстоянии Левенштейна, добавлена операция транспозиции (перестановки) символов. Фредерик Дамерау показал, что 80 % ошибок при наборе текста человеком являются транспозициями. Из приведенного выше анализа известных методов поиска строк становится ясным, почему именно метрика Левенштейна легла в основу разработанного авторами метода нечеткого поиска реквизитов физических лиц в базах данных.

3. Математическая модель

Рассмотрим общую метрику Левенштейна, которая поддерживает три операции со строкой: вставки, замены и удаления символа, причем все три операции имеют одинаковый вес [8-9]. Для дальнейшей работы была введена лингвистическая переменная «схожесть строк». Решено выделить следующие термы: «строки совпадают», «строки почти совпадают», «строки похожи», «строки и похожи и непохожи одновременно», «строки не похожи».

В результате анализа функций принадлежности лингвистических термов возникла необходимость модификации метода вычисления метрики Левенштейна. Потребовалось модифицировать метрику таким образом, чтобы расстояние между строками зависело, в том числе, и от длины сравниваемых строк.

Теорема. Обозначим при помощи величины $p(s_1, s_2)$ метрику Левенштейна, а величиной $\|s_i\|$ — длину строки s_i . Тогда функция:

$$r(s_1, s_2) = \frac{p(s_1, s_2)}{\max \{\|s_1\|, \|s_2\|\}}, \quad (1)$$

является метрикой.

Доказательство. Поскольку $p(s_1, s_2)$ — метрика, то имеем:

$$p(s_1, s_2) \geq 0,$$

$$p(s_1, s_2) = p(s_2, s_1),$$

$$p(s_1, s_2) + p(s_2, s_3) \geq p(s_1, s_3)$$

для любых строк s_1, s_2 и s_3 .

Учитывая эти соотношения и равенство (1), приходим к выводу, что $r(s_1, s_2)$ удовлетворяет первым двум аксиомам, определяющим метрику. Остается доказать, что для любых строк s_1, s_2 и s_3 функция $r(s_1, s_2)$ удовлетворяет неравенству треугольника:

$$r(s_1, s_2) + r(s_2, s_3) \geq r(s_1, s_3).$$

Запишем это неравенство в виде:

$$\frac{p(s_1, s_2)}{\max \{\|s_1\|, \|s_2\|\}} + \frac{p(s_2, s_3)}{\max \{\|s_2\|, \|s_3\|\}} - \frac{p(s_1, s_3)}{\max \{\|s_1\|, \|s_3\|\}} \geq 0.$$

Возможны следующие случаи:

1. $\|s_1\| \leq \|s_2\| \leq \|s_3\|$
2. $\|s_2\| \leq \|s_3\| \leq \|s_1\|$
3. $\|s_3\| \leq \|s_1\| \leq \|s_2\|$
4. $\|s_2\| \leq \|s_1\| \leq \|s_3\|$
5. $\|s_1\| \leq \|s_3\| \leq \|s_2\|$
6. $\|s_3\| \leq \|s_2\| \leq \|s_1\|$

Рассмотрим первый случай. Имеем:

$$\begin{aligned} & \frac{p(s_1, s_2)}{\max \{\|s_1\|, \|s_2\|\}} + \frac{p(s_2, s_3)}{\max \{\|s_2\|, \|s_3\|\}} - \frac{p(s_1, s_3)}{\max \{\|s_1\|, \|s_3\|\}} = \frac{p(s_1, s_2)}{\|s_2\|} + \frac{p(s_2, s_3)}{\|s_3\|} - \frac{p(s_1, s_3)}{\|s_3\|} \geq \\ & \geq \frac{1}{\|s_3\|} (p(s_1, s_2) + p(s_2, s_3) - p(s_1, s_3)) \geq 0. \end{aligned}$$

Таким образом, для первого случая неравенство треугольника выполняется. Поскольку второй случай аналогичен первому, на основании подобных выкладок делаем вывод, что для второго случая неравенство треугольника также выполняется.

Перейдем к рассмотрению третьего случая. Итак, в третьем случае имеем:

$$r(s_1, s_2) + r(s_2, s_3) - r(s_1, s_3) = \frac{1}{\|s_2\|} (r(s_1, s_2) + r(s_2, s_3)) - \frac{1}{\|s_1\|} r(s_1, s_3)). \quad (2)$$

Рассмотрим вопрос о том, когда достигается минимум функции, находящейся в правой части этого равенства. Понятно, что если выражение $r(s_1, s_2) + r(s_2, s_3)$ достигает минимума, а $r(s_1, s_3)$ максимума, то значение всего выражения будет минимальным. Указанные два условия могут выполняться одновременно, если одновременно выполняются два следующих утверждения:

1. строки s_1 и s_3 не имеют общих символов;
2. строки s_1 и s_3 входят в качестве подстрок в s_2 .

Тогда:

$$r(s_1, s_3) = \max\{\|s_1\|, \|s_3\|\} = \|s_1\|,$$

$$r(s_1, s_2) = \|s_3\| + \|C\|, \quad r(s_2, s_3) = \|s_1\| + \|C\|,$$

и, таким образом, минимальное значение выражения (2) запишется в виде:

$$\frac{\|s_3\| + \|C\| + \|s_1\| + \|C\|}{\|s_3\| + \|s_1\| + \|C\|} - \frac{\|s_1\|}{\|s_1\|} = \frac{\|C\|}{\|s_3\| + \|s_1\| + \|C\|} \geq 0.$$

Следовательно, в третьем случае для функции $r(s_1, s_3)$ также выполняется неравенство треугольника. Остальные случаи аналогичны уже рассмотренным. Таким образом, функция $r(s_1, s_2)$ является метрикой, заданной на множестве строк. Теорема доказана.

Замечание. Функция $r(s_1, s_2)$ принадлежит отрезку $[0, 1]$ для любых строк s_1 и s_2 .

В предложенном алгоритме данная метрика применяется для работы со строковыми реквизитами физических лиц, к которым относятся ФИО, адрес, документ и т.д. В связи с этим построенная с использованием данной метрики лингвистическая переменная позволяет обрабатывать запросы поиска для человека, похожего на другого человека по реквизитам. Приняв от пользователя такой запрос, мы фактически получаем два значения: значение искомого реквизита и радиус поиска.

4. Алгоритм нечеткого поиска реквизитов физических лиц

Укрупненная блок-схема разработанного алгоритма нечеткого поиска реквизитов физических лиц в базах данных с учетом возможных неточностей или ошибок ввода. С помощью указанных процедур программа формирует набор, представлен на рис. 1.

В реализации алгоритма на языке PL-SQL СУБД Oracle 11g за предварительную выборку всех записей, отдаленно похожих на искомую, отвечает блок «Запрос количества идентичных людей в базе данных». Этот блок работает по алгоритму прямого частичного сравнения разных наборов реквизитов, например, имени, отчества и даты рождения, формируя тем самым рабочей набор данных для рассматриваемого алгоритма идентификации. Затем в работу вступает «Блок сравнения реквизитов», ключевые функции которого отводятся логически выделенным процедурам COMPARISON_STRING и COMPARISON_NUMBER, созданным на основе модифицированного метода вычисления метрики Левенштейна, которые позволяют проводить интеллектуальное сравнение двух похожих строк или чисел, с совпадений, и по результатам обработки предлагаемой и искомой записи выносит решение об идентичности строк. Например, у человека совпадает имя, отчество, дата рождения, и номер паспорта, а в фамилии допущена ошибка в одну букву. В данном случае программа однозначно идентифицирует реквизиты. Данные процедуры могут применяться не только для идентификации реквизитов, но также везде, где требуется полнотекстовый поиск с нечетко заданными входными данными.

Алгоритм идентификации аккумулирует так называемый «опыт прошлых идентификаций» и записывает его в специально отведенное место в базе данных, для использования в последующих идентификациях. Это позволяет сохранить не только результаты автоматической работы программы, но и решения операторов после отработки ими оставшихся не найденных реквизитов.

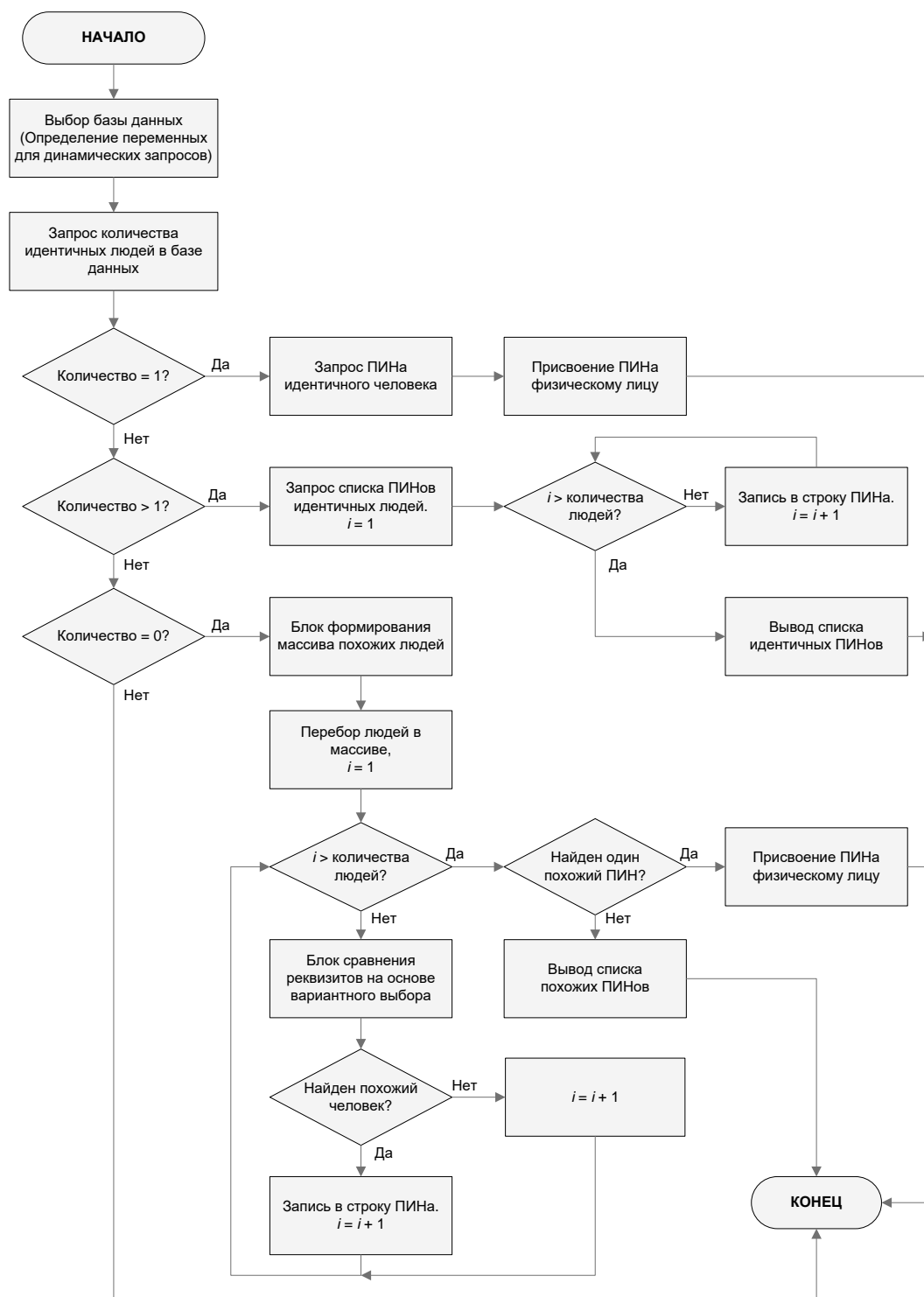


Рис. 1. Укрупненная блок-схема алгоритма нечеткого поиска реквизитов физических лиц в базах данных.

5. Заключение

Рассмотренный метод нечеткого поиска персональных данных позволяет быстро определять людей, используя данные ранее проведенного поиска. Встроенная система приоритета реквизитов позволяет идентифицировать человека в таких случаях, как смена фамилии, имени, переезд, ошибки при ручном вводе данных, а также при частично отсутствующих реквизитах.

В перспективе данный алгоритм обладает возможностью успешного внедрения в системы глобального объединения хранилищ государственных или коммерческих организаций, для ведения единой базы данных населения любой страны мира и т.д. Логическая структура разработанного алгоритма позволяет реализовать его на любом популярном языке программирования. Масштабируемость алгоритма дает возможность применять программные процедуры на его основе как в малых организациях, так и в крупных корпорациях, везде, где ведется и актуализируется реестр данных физических лиц. Возможные примеры использования: портал госуслуг, медицинские электронные системы, кадровые и бухгалтерские системы учета служащих, банковские системы хранения данных о клиентах и т.п.

Алгоритм реализован на языке PL-SQL системы управления базами данных Oracle 11g. Разработанное программное обеспечение, реализующее метод нечеткого поиска персональных данных, внедрено и успешно функционирует в ряде муниципальных и государственных учреждений г. Тольятти Самарской области. В 2016 году программное обеспечение, включающее в себя реализацию разработанного алгоритма, внедрено в информационную систему по исполнению бюджета г. Тольятти.

Литература

- [1] Международный фонд автоматической идентификации. Технологии автоматической идентификации [Электронный ресурс]. – Режим доступа: <http://www.fond-ai.ru/art1/art223.html>, свободный. Яз. рус. (дата обращения 28.01.2017).
- [2] Желудков, А. В., Макаров, Д. В., Фадеев, П. В. Особенности алгоритмов нечёткого поиска. Москва, Инженерный вестник МГТУ им. Н.Э. Баумана, 2014. – С. 502-503
- [3] Soundex метод нечёткого поиска, URL: <https://ru.wikipedia.org/wiki/Soundex> (дата обращения 28.01.2017)
- [4] Харитоненков, А.В. «Поиск на неточное соответствие: коды Хемминга», <http://www.jurnal.org/articles/2009/inf32.html> (дата обращения 28.01.2017)
- [5] Двоичный алгоритм поиска подстроки, URL: https://ru.wikipedia.org/wiki/Двоичный_алгоритм_поиска_подстроки (дата обращения 28.01.2017)
- [6] Задача о редакционном расстоянии, алгоритм Вагнера-Фишера, URL: http://neerc.ifmo.ru/wiki/index.php?title=Задача_о_редакционном_расстоянии_алгоритм_Вагнера-Фишера (дата обращения 28.01.2017)
- [7] Расстояние Дамерау — Левенштейна, URL: https://ru.wikipedia.org/wiki/Расстояние_Дамерау_—_Левенштейна (дата обращения 28.01.2017)
- [8] Левенштейн, В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии наук СССР. – 1965. – Т. 163. – № 4. – С. 845–848.
- [9] Бойцов, Л.М. Анализ строк [Электронный ресурс]. – Режим доступа: http://itman.narod.ru/articles/infoscope/string_search.1-3.html, свободный. Яз. рус. (дата обращения 28.01.2017).